**Class: AccountTransaction**

* Method: updateAddCredit(String data, ArrayList<String> usersData)
  + TestCase#1:

**Description - newseller adds 100 credits to his account**

* + - data = “06 newseller SS 000100.00”
    - userData = {“admin AA 543278.23”,

“newseller SS 005201.25”

“event host FS 020850.00” }

Result:

actualUserData = {“admin AA 543278.23”,

“newseller SS 005301.25”

“event host FS 020850.00” }

* Method: updateRefund(String data, ArrayList<String> usersData)
  + TestCase#1:

**Description – newseller refunds 100 credits to newbuyer**

* + - data = “05 newbuyer newseller 000100.00”
    - userData = { “admin AA 543278.23”,

“newseller SS 005201.25”,

“event host FS 020850.00”

“newbuyer BS 000500.00” }

Result:

actualUserData = {“admin AA 543278.23”,

“newseller SS 005101.25”

“event host FS 020850.00”

“newbuyer FS 000600.00” }

* Method: updateCreate(String data, ArrayList<String> usersData)
  + TestCase#1:

**Description – a new standard user is created**

* + - data = “01 standard FS 000000.00”
    - userData = { “admin AA 543278.23”,

“newseller SS 005201.25”,

“event host FS 020850.00” }

Result:

actualUserData = {“admin AA 543278.23”,

“newseller SS 005201.25”

“event host FS 020850.00”

“standard FS 000000.00” }

**Class: TicketTransaction**

* Method: updateDelete(String data, ArrayList<String> usersData, ArrayList<String> ticketData)
  + TestCase#1 [Has no outstanding tickets]:

**Description – delete newseller without any outstanding tickets**

*80% Statement Coverage*

* + - data = “02 newseller SS 005201.25”
    - userData = { “admin AA 543278.23”,

“newseller SS 005201.25”

“event host FS 020850.00” }

* + - ticketData =

{ “Test Event event host 015 012.45” }

Result:

actualUserData = {“admin AA 543278.23”,

“event host FS 020850.25” }

* + TestCase#2 [Has outstanding tickets]:

**Description – delete newseller with available tickets**

*100% Statement Coverage*

* + - data = “02 newseller SS 005201.25”
    - userData = { “admin AA 543278.23”,

“newseller SS 005201.25”

“event host FS 020850.00” }

* + - ticketData =

{ “Test Event event host 015 012.45”

“NymH Event newseller 055 019.99” }

Result:

actualUserData = {“admin AA 543278.23”,

“event host FS 020850.00” }

actualTicketData = { “Test Event event host 015 012.45” }

* Method: updateSell(String data, ArrayList<String> ticketData)
  + TestCase#1:

**Description – add admin event title by admin, event to ticket array**

* + - data = “03 admin event title admin 012 005.00”
    - ticketData =

{ Test Event event host 015 012.45 }

* Method: updateBuy(String data, ArrayList<String> usersData,

ArrayList<String> ticketData, ArrayList<String> transactionData)

* + TestCase#1 [Ticket Does Not Exist]:

**Description – admin attempts to buy LCS Series ticket that does not exist**

*~ 20.5% Statement Coverage*

* + - data = “04 LCS Series newseller 002 032.95”
    - userData ={ “admin AA 543278.23”,

“newseller SS 005201.25”,

“event host FS 020850.00” }

* + - ticketData =
* { Test Event newseller 015 012.45 }
  + - transactionData =

{ “04 LCS Series newseller 002 032.95”,

“00 admin AA 543278.23” }

* + TestCase#2 [Ticket Does Exist]:

**Description – admin buys 3 test event tickets from event host**

*~ 97% Statement Coverage*

* + - data = “04 Test Event event host 012 012.45”
    - userData ={ “admin AA 543278.23”,

“newseller SS 005201.25”,

“event host FS 020850.00” }

* + - ticketData =
* { Test Event event host 015 012.45 }
  + - transactionData =

{ “04 Test Event event host 012 012.45”,

“00 admin AA 543278.23” }

**Class: BackEnd**

The file not found exception for the 3 methods below can only occur if the data files are not in the directory specified. They do not actually take any input paths, the paths are already initialized as private, final variables. The manual removal of files from directory, test was conducted and they were successful in displaying file not found error message!

* Method: ArrayList<String> getTicketData()

**Description – load Ticket Data from Tickets File into an array**

*100% Statement Coverage*

* Method: ArrayList<String> getUsersData()

**Description – load Users Data from Users File into an array**

*100% Statement Coverage*

* Method: ArrayList<String> getTransactionData()

**Description – load Transaction Data from Transaction File into an array**

*100% Statement Coverage*

* Method: findUserIndex(String target, ArrayList<String> data)
  + TestCase#1 [User Exists]

**Description – call function to find index of existing target user in user ArrayList**

*80% Statement Coverage*

* + - target = “admin ”
    - data = { “admin AA 543278.23”,

“newseller SS 005201.25”,

“event host FS 020850.00” }

* + TestCase#2 [User Does Not Exists]

**Description – attempt to find index of non-existing user in user ArrayL**

*80% Statement Coverage*

* + - target = “admin ”
    - data = { “admin AA 543278.23”,

“newseller SS 005201.25”,

“event host FS 020850.00” }

* Method: findBuyer(ArrayList<String> data, intstartIndex)
  + TestCase#1 [Buyer Exists]

**Description – call function to find buyer at given index in array**

*80% Statement Coverage*

* + - data = { “00 buyer BS 999999.99”,

“02 event host SS 000000.00” }

* + - startIndex = 0
  + TestCase#2 [Buyer not found – **NOT POSSIBLE**, Buy order must follow logout]

**Description – attempt to find buyer at given index in array**

*80% Statement Coverage*

* + - data = { “01 Jude AA 000000.00”,

“02 event host SS 000000.00” }

* + - startIndex = 0
* Method: saveToOutput(ArrayList<String> usersData, ArrayList<String> ticketData)
  + TestCase#1

**Description – save modified usersData and ticketData from arrays to respective files.**

* + - userData = { “admin AA 543278.23”,

“newseller SS 005201.25”

“event host FS 020850.00” }

* + - ticketData =

{ “Test Event event host 015 012.45” }

--------------------------------------------------------------------------------------

* **(Loop Coverage / Branch Coverage)**
* Method: findBuyer(ArrayList<String> data, int startIndex)
  + TestCase#0 [Goes through for loop zero times ]:

**Description – No User Data**

* + - userData = {}
  + TestCase#1 [Goes through for loop 1 time]:

**Description – Only one User data and it starts with “00”**

* + - userData = {“ 00 buyer BS 999999.99”}
  + TestCase#2 [Goes through for loop 2 times, passes if Statement]:

**Description – Goes through list and finds buyer**

* + - userData = { "00 buyer BS 999999.99",

"02 event host SS 000000.00"}

* + TestCase#3 [Goes through for loop many times , passes if statement]:

**Description – Finds buyer in a big list**

* + - userData = { “00 buyer BS 999999.99”,

“02 event host SS 000000.00”,

“01 Jude AA 000000.00”,

“01 Matt AA 000000.00” }

* + TestCase#4 [If statement never passes]:

**Description – not found**

* + - userData = { “01 Jude AA 000000.00”,

“02 event host SS 000000.00”}

--------------------------------------------------------------------------------------